

ANSWERS



- 1) Get all account records with fields id and name with industry field value as Energy.

Concept: SOQL Query with where clause.

```
select id,name from account where industry='Energy'
```

- 2) Get 10 account records with fields id and name.

Concept: SOQL Query using limit.

```
select id,name from account limit 10
```

- 3) Get all contact records with fields id, last name, and account name that has account industry field value as Energy.

Concept: SOQL Query with where clause. Cross object field to display the account fields from contact object.

```
select id,lastname,account.name from contact where account.industry='Energy'
```

- 4) Create a list that can hold the integers with 1,2,3 as values.

Concept: list collection

```
1. List<Integer> lstIntNumbers= new List<Integer>();
2. lstIntNumbers.add(1);
3. lstIntNumbers.add(2);
4. lstIntNumbers.add(3);
```

- 5) Create a list that to hold your friends' names.

Concept: list collection

```
1. List<String> lstStrName= new List<String>();
2. lstStrName.add('Sam');
3. lstStrName.add('Jack');
4. lstStrName.add('Veeru');
```

- 6) Create a set to hold the integers with 1,2,4,5,1 as values. Check the size of the set.

Concept: set collection

```
1. set<Integer> setIntNumbers= new set<Integer>();
2. setIntNumbers.add(1);
3. setIntNumbers.add(2);
4. setIntNumbers.add(4);
5. setIntNumbers.add(5);
6. setIntNumbers.add(1);
```

- 7) Create a map to hold your friend's phone number and names. Remember to keep the key unique.

Concept: Map collection

```
1. //Phone number is unique, so keep it as key. Feel free to add a country code too.
2. //to all country code, I had taken the key data type as string.
3. map<String,string> phoneNumberToName = new map<String,string>();
4. phoneNumberToName.put('999XXXXXX', 'Tom');
5. phoneNumberToName.put('987XXXXXX', 'Jerry');
```

8) Add all the account records which has industry as Energy to a list collection and print the size of the list.

Concept: list collection, SOQL query, for-each loop and system debug

```
1. List<account> lstaccount= new List<account>();
2. for(account a :[select id,name from account where industry='Energy']){
3.     lstaccount.add(a);
4. }
5. system.debug('size of the lstaccount is '+lstaccount.size()); // size of the
    lstaccount is 3 (results may vary)
```

9) Print all the account records names which had industry as Energy.

Concept: SOQL query, for-each loop and system debug

```
1. for(account a :[select id,name from account where industry='Energy']){
2.     system.debug(a.name);
3. }
```

10) Take the list from step 1 and then to a new list.

Concept: list collection and for-each loop

```
1. List<account> lstNewAccount= new List<account>();
2. List<account> lstaccount= new List<account>();
3. for(account a :[select id,name from account where industry='Energy']){
4.     lstaccount.add(a);
5. }
6. // learning how to add items from one list to another.
7. for(account acc : lstaccount){
8.     lstNewAccount.add(acc);
9. }
```

11) Add all the case records which has account industry as Energy to a list collection only if 'escalated' field on case is false and print the size of the list too.

Concept: list collection, for-each loop, if and system.debug.

```
1. List<case> lstcase= new List<case>();
2. for(case c :[select id,IsEscalated from case where account.industry='Energy']){
3.     //adding if condition to check if the donotcall is false.
4.     if(c.IsEscalated==false){
5.         lstcase.add(c);
6.     }
7. }
8. system.debug('Energy industry cases which are not escalated '+lstcase.size());
```

12) Add all the case records which has account industry as Energy into different lists based on 'escalated' field values and print the sizes of the lists too.

Concept: list collection, for-each loop, if-else and system.debug.

```
1. List<case> lstNotEscalatedcase= new List<case>();
2. List<case> lstEscalatedcase= new List<case>();
3. for(case c :[select id,IsEscalated from case where account.industry='Energy']){
4.     //adding if condition to check if the donotcall is false.
5.     if(c.IsEscalated==false){ // //if(!c.IsEscalated){ is another way of writing
        this if condition.
6.         lstNotEscalatedcase.add(c);
```

ANSWERS

```
7.     }else{
8.         lstEscalatedcase.add(c);
9.     }
10. }
11. system.debug('Energy industry cases which are not escalated
    '+lstNotEscalatedcase.size());
12. system.debug('Energy industry cases which are escalated
    '+lstEscalatedcase.size());
```

13) Add all the case records into different lists based on the status field values and print the size of the list.
Concept: list collection, for-each loop, if, else if and else, and system.debug.

```
1. List<case> lstNewcase= new List<case>();
2. List<case> lstWorkingcase= new List<case>();
3. List<case> lstClosedcase= new List<case>();
4. List<case> lstEscalatedcase= new List<case>();
5. List<case> lstOthercase= new List<case>();
6.
7. for(case c :[select id,status from case]){
8.     //adding if condition to check if the case status is new.
9.     if(c.status.equalsIgnoreCase('New')){ // //if(!c.IsEscalated){ is another
    way of writing this if condition.
10.         lstNewcase.add(c);
11.     }
12.     // check if the case status is working.
13.     else if(c.status.equalsIgnoreCase('Working') ){
14.         lstWorkingcase.add(c);
15.     }
16.     // check if the case status is Closed.
17.     else if(c.status.equalsIgnoreCase('Closed') ){
18.         lstClosedcase.add(c);
19.     }
20.     // check if the case status is Escalated.
21.     else if(c.status.equalsIgnoreCase('Escalated') ){
22.         lstEscalatedcase.add(c);
23.     }
24.     else{
25.         lstOthercase.add(c);
26.     }
27. }
28. system.debug('number of new cases are: '+lstNewcase.size());
29. system.debug('number of Working cases are: '+lstWorkingcase.size());
30. system.debug('number of Closed cases are: '+lstClosedcase.size());
31. system.debug('number of Escalated cases are: '+lstEscalatedcase.size());
32. system.debug('number of Other cases are: '+lstOthercase.size());
```

14) Create a map with to hold the case records which has account industry as Energy. Print the size of the map. Try using the containskey and keyset methods.

Simple version:

Concept: Map collection, for-each loop and system.debug.

```
1. // map - key id as it is unique.
2. // value - case record.
3. map<id,case> IdTocase = new map<id,case>();
4. for(case c:[select id,caseNumber from case where account.industry='Energy']){
5.     IdTocase.put(c.id,c);
6. }
7. system.debug('size of the map IdTocase is '+IdTocase.size());
```

ANSWERS

adding containskey and keyset to the code.

```
1. // map - key id as it is unique.
2. // value - case record.
3. map<id,case> IdTocase = new map<id,case>();
4. for(case c:[select id,caseNumber from case where account.industry='Energy']){
5.     //adding a if condition with containskey to show how to add the records to map
    only if they are not present in it.
6.     // in this scenario this check is not needed. we are adding it for practicing
    containskey concept.
7.     if(!IdTocase.containskey(c.id)){
8.         IdTocase.put(c.id,c);
9.     }
10.
11. }
12. system.debug('size of the map IdTocase is '+IdTocase.size());
13. system.debug('key in map IdTocase are '+IdTocase.keySet());
```

15) Create account with your fav brand name using apex.

Concept: Insert DML.

```
1. account a = new account();
2. a.name='salesforce';
3. insert a;
```

16) Create a contact with your last name and attach it to the account created on step 1 using apex.

Concept: SOQL Query, Insert DML.

```
1. //getting the account created in step 1. adding limit 1 to limit any duplicate
    records with same name.
2. account a = [select id, name from account where name='salesforce' limit 1];
3. //creating contact
4. contact c = new contact();
5. c.lastname='astro';
6. //populate the account field.
7. c.accountid=a.id;
8. insert c;
```

version 2

```
1. //Creating contact.
2. contact c = new contact();
3. c.lastname='astro';
4. //populate account id
5. c.accountid=[select id from account where name='salesforce'][0].id;
6. //using insert dml on the contact c variable
7. insert c;
```

17) Update the first name field on contact in step 2.

Concept: SOQL Query, Update DML

```
1. //getting the contact created in step 2. adding limit 1 to limit any duplicate records
    with same last name.
2. contact c = [select id, lastname, firstname from contact where lastname='astro' limit
    1];
3. c.firstname='cool';
```

4. update c;

18)append all account records description field with apex test update.

Concept: SOQL Query, Update DML, List collection and if-else

```
1. List<account> lstAccount= new List<account>();
2. for(account a :[select id,name,Description from account where name ='salesforce']){
3.     //appending the apex test update to existing account description.
4.     //if the description is present on the record, append the apex test upate to
    existing desc
5.     if(!String.isBlank(a.Description)){
6.         a.Description+='apex test update';
7.     }
8.     //directly assign apex test update to the desc field if the field is blank.
9.     else{
10.        a.Description='apex test update';
11.    }
12.    //adding to the account records to list lstAccount for bulkification
13.    lstAccount.add(a);
14. }
15.
16. //check if lstaccount is not empty
17. if(lstaccount!=null && !lstaccount.isEmpty()){
18.     update lstAccount;
19. }
```